# Secure Position and Time Information by Server Side PRS Snapshot Processing

Alexander Rügamer, Daniel Rubino, Ivana Lukčin, Simon Taschke, Manuel Stahl, Wolfgang Felber
*Fraunhofer Institute for Integrated Circuits IIS, Nuremberg, Germany*

## BIOGRAPHY

Alexander Rügamer received his Dipl.-Ing. (FH) degree in Electrical Engineering from the University of Applied Sciences Wuerzburg-Schweinfurt, Germany, in 2007. Since then he has been working at the Fraunhofer Institute for Integrated Circuits IIS in the Field of GNSS receiver development. He was promoted to Senior Engineer in February 2012. Since April 2013, he is head of a research group dealing with secure GNSS receivers and receivers for special applications. His main research interests focus on GNSS multi-band reception, integrated circuits and immunity to interference.

Daniel Rubino received his MSc. degree in Industrial Engineering and Management from the University Erlangen-Nuremberg, Germany, in 2015. Upon graduation he joined the Fraunhofer Institute for Integrated Circuits IIS, where he is involved in signal processing for GNSS receiver including software and hardware development.

Ivana Lukčin received her MSc. degree in Mathematics from the University of Zagreb, Croatia, in 2012. Since then, she has been working at the Fraunhofer Institute for Integrated Circuits IIS in the field of compressed sensing and snapshot positioning.

Simon Taschke received his Msc. degree in Geodesy and Geoinformatics from the University of Stuttgart, Germany, in 2015. Since the same year he has been working at the Fraunhofer Institute for Integrated Circuits IIS in the field of GNSS receiver development and testing.

Manuel Stahl received his Dipl.-Inf. degree in Computer Sciences from the University of Wuerzburg, Germany, in 2009. Since the same year he has been working at the Fraunhofer Institute for Integrated Circuits IIS in the field of GNSS receiver software development for embedded systems and mobile robotics.

Wolfgang Felber received his Dipl.-Ing. degree in electrical engineering in 2002 and his doctoral degree Dr.-Ing. in 2006 from Helmut-Schmidt-University of Federal Armed Forces Hamburg, Germany. Since 2014 he is head of the Power Efficient Systems department of Fraunhofer IIS in Nuremberg. The main topics in his department are energy harvesting and low power technologies, hardware development of satellite navigation receivers and sensor fusion in positioning applications.

## ABSTRACT

An alternative way to benefit from spoofing-resistant Galileo Public Regulated Service (PRS) signals is to use a server side processing concept, where all security-related processing steps are outsourced. Having outlined the architecture with user terminal, communication channel and remote server, the snapshot positioning algorithms with different pseudorange reconstruction methods are presented. To reduce the snapshot size, data compression methods are possible, however, degrading the effective $C/N_0$. The results are demonstrated, processing E1 Pseudo-PRS signals from an RF constellation simulator and applying different compression techniques. Multiple raw data snapshots, with sizes from 810 to 10 kByte are evaluated providing Galileo E1 PRS snapshot positioning $1\sigma$ accuracy from 1.1 to 6.0 m. Finally, a method to estimate the accuracy from a single snapshot result is outlined.

## INTRODUCTION

Galileo Public Regulated Service (PRS) is a special, cryptographically protected satellite navigation service intended for government authorized users. The access to PRS is regulated by decision No 1104/2011/EU of the European Parliament and of the Council and controlled by member states.

With the encryption used in Galileo PRS, the manipulation of a PRS-obtained time and position is hardly possible. This allows the realization of many demanding or security critical applications that could not be operated using conventional GPS or Galileo Global Navigation Satellite Services (GNSS).

However, PRS receivers won't be available to ordinary users. Moreover, in general, PRS receivers will never be as inexpensive as standard open service (OS) receivers and are

more complex in their handling due to their security module and the necessary key management.

An alternative to these conventional PRS receivers are server-based or remote processing PRS receivers [1]: The general idea is to outsource the PRS signal processing to a secure server environment. The user terminal itself samples only raw data signals and forwards them to the server, where the PRS information included in the raw samples is processed.

The PRS is only used in a passive, non-real-time way by the user. The remote PRS server service provider decides if and which information is returned to the user. This leads to certain advantages over conventional Galileo PRS receivers: Firstly, users do not have to care about the security requirements of PRS since there are no PRS security-related functions on the user receiver side. Secondly, the Galileo PRS security cannot in any way be endangered by the user, since the user terminal does not contain any PRS relevant methods or information (e.g. a security module).

Having outlined the Galileo signals, the architecture of a server side PRS processing concept with user terminals is described. Then we discuss methods to obtain a position and time information out of raw snapshot data. We start by describing the state of the art of snapshot processing with its requirements and limitations. After that, we introduce a new snapshot position technique that directly resolves the acquisition code phase ambiguity by exploiting pilot signals' secondary code resulting in unambiguous measurements. Finally, we show how the snapshot techniques can be applied to Galileo PRS signals. The snapshot size generated by the user terminal is crucial for most remote processing applications. Therefore, data compression methods and their impacts on the probability of detection and accuracy estimation are explained. The experimental setup to record and evaluate GNSS RF simulator E1 Pseudo-PRS signals is outlined and the snapshot processing results using different compression setups are presented. Having discussed a method with proof of concept results to estimate the accuracy from a single snapshot, the paper concludes with the outline of possible applications for server-side PRS snapshot positioning.

## GALILEO PRS

The European GNSS Galileo provides three different global navigation services: Open Service (OS), Commercial Service (CS), and the Public Regulated Service (PRS). Galileo OS is similar to the free services of GPS and GLONASS. The Galileo CS is not yet fully defined. Galileo PRS features two encrypted signals on two frequency bands and targets government authorized users, e.g. police, border control, emergency, armed forces, Search and Rescue, and also operators of critical infrastructures like telecommunication and energy networks as well as critical transports.

As shown in Fig. 1, the Galileo PRS signals are transmitted in a coherent way together with the OS and CS signals over the E1A and E6A frequency bands, using a binary offset carrier (BOC) modulation denoted as BOCc(15,2.5) and BOCc(10,5), respectively. BOCc uses a cosine phased subcarrier resulting in higher frequency components than a sine phased subcarrier used in BOCs modulations of e.g. Galileo E1BC OS. As a result, more energy is shifted to the edges of the band. This improves the spectral separation with the coexisting OS and CS signals and the theoretical tracking performance [2].

Thanks to the encryption used, Galileo PRS can add a legal value on its PRS position, velocity and time (PVT) solution since anti-spoofing is guaranteed. This property is a key opener to many critical and demanding applications mostly in security-related areas. Although everyone can receive the PRS raw data, only someone having the decryption key is capable of generating the PRS pseudo-random-noise (PRN) sequences to despread the PRS signals and to process their messages.

## GALILEO OS/PRS SNAPSHOT RECEIVER

### *Architecture*

The concept of a PRS sample and processing snapshot receiver is to outsource the actual acquisition and position calculation to a secure server environment. As illustrated in Fig. 2, the architecture consists of a number of user terminals and a PRS remote processing server. The authorized users can benefit from the PRS without jeopardizing Galileo PRS security. The PRS remote processing server alone supports all the PRS security processing features, whereas inexpensive and miniaturized end user devices are feasible as user terminals without an integrated PRS security module.

### *Antenna and Data Grabber*

The user terminal receives OS/PRS signals from a PRS suitable antenna on either E1A, E6A, or both frequency bands. Afterwards, a raw OS/PRS data snapshot is recorded using a front-end also denoted as data grabber. The recorded raw data snapshot can be reduced in size according to the application requirements.

To be able to receive the BOCc modulated PRS signals, a reception bandwidth of approx. 40 MHz is required with respect to the E1 and E6 center frequencies of 1575.42 MHz and 1278.75 MHz. Depending on the intended application, a single band reception of E1 or E6, or even the reception of a single main lobe of the BOC signal only might be sufficient. However, it should be noted that the dual-band capability of PRS provides a significantly higher jamming protection: If one signal band is jammed, the other one might still be usable. Moreover, limiting the reception to a single sideband of the BOC-modulated PRS signal will add some $C/N_0$ reception power loss, as discussed in the "Data Compression Effects" section.

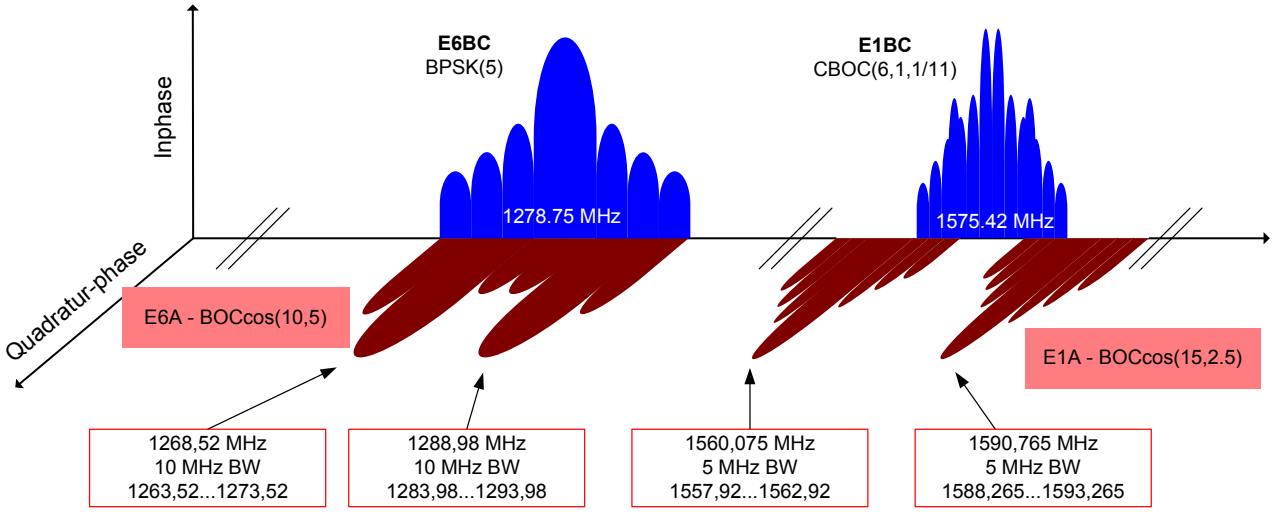



**Figure 1.** Galileo PRS signals

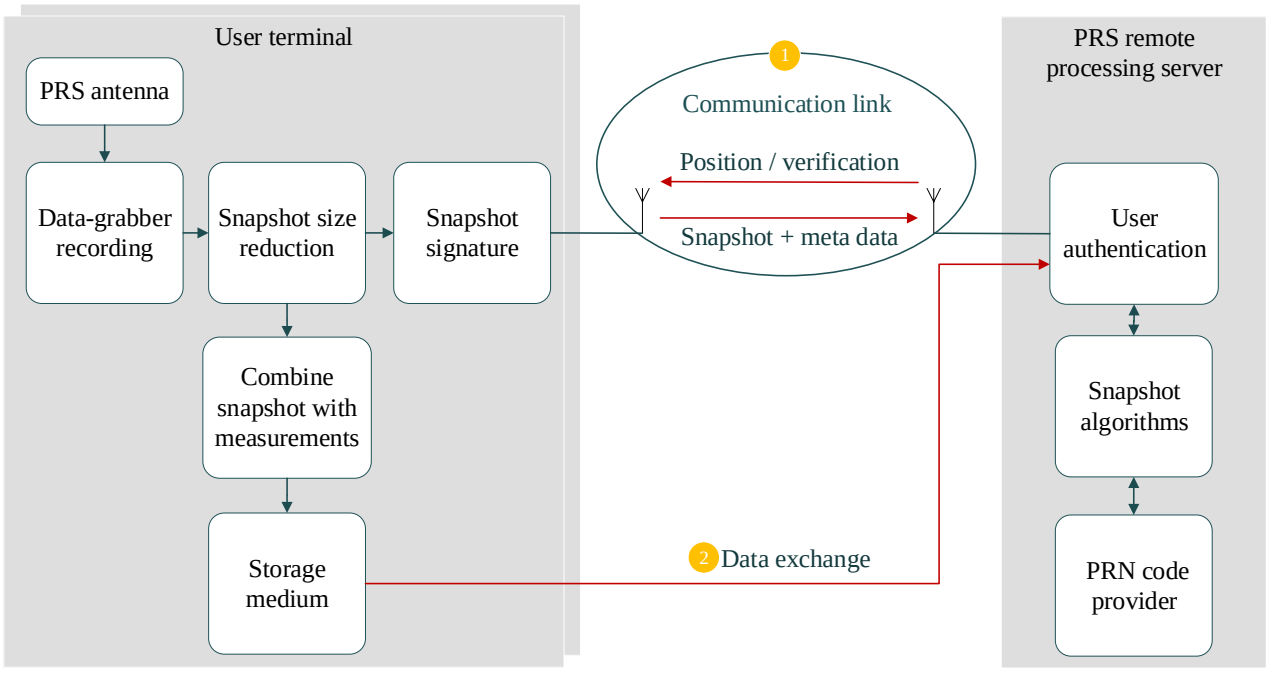



**Figure 2.** System architecture with communication link (1) and data transfer link for postprocessing (2)

Since for most applications the user terminal should be small and inexpensive, the wideband signals on two frequency bands are a challenge. There are some geodetic GNSS antennas capable of receiving the dual-band BOCc modulated PRS signals, however, they are by far neither inexpensive nor small. Most mobile end user devices use integrated antennas or chip antennas. These antennas normally only feature the reception of the E1 frequency band, and are often limited to GPS L1 C/A only using narrow, integrated filters. In general, miniaturized antennas which can receive GPS L1 together with the G1 GLONASS signals centered around 1602 MHz are suited to at least also receive the upper Galileo PRS E1 sidelobe. Another downside of these integrated antennas is their limited efficiency and linear polarization only leading to additional $C/N_0$ reception losses.

Similar restrictions hold for the data grabber. Commercial GPS RF front-end chips like the *Maxim MAX2769* [3] can be tuned in the frequency range of E1 from 1550 MHz to 1610 MHz, but their overall reception bandwidth is limited to approx. 18 MHz dual-sided bandwidth. This is not enough to receive the full BOCc E1A signal but can be sufficient to receive E1 OS together with either the upper or lower E1A sidelobe.

*Snapshot Size Reduction*

In general, the user terminals' raw data snapshot size is determined by the complex sampling rate $f_s$, quantization bits $Q$, and snapshot length $l$ with:

$$size = f_s \cdot Q \cdot l \tag{1}$$

Data compression can be applied in terms of resampling and filtering the received signals (to reduce the $f_s$), reducing the quantization $Q$ or limiting the recording length $l$ or size in other ways.

For many applications, the trade-offs between snapshot-size, probability of detection (reliability), and accuracy is crucial. The different data compression effects are discussed in the respective section.

*Combination of Snapshot with a Measurement / Snapshot Signature*

The raw PRS snapshot can be used as a digital fingerprint on a measurement, file, or document to be authenticated with a time and/or georeference.

Depending on the intended application and the required security level, the raw sample snapshot device should be encapsulated together with the actual measurement device in a tamper-proof housing. Such protected units are already standard for many devices like on-board units, electricity smart meters, etc.

The goal is to cryptographically combine the raw PRS snapshot with the measurement to be signed in a way that the positioning and time information of this measurement can be authenticated using PRS later on. One way of doing this combination is using a public/private key infrastructure. It is assumed that the user terminal incorporates a private key and has shared its public key with the customer. The private key is used to sign a hash value of both the raw PRS snapshot and the measurement. The public key, hash values, and the actual measurement with raw PRS snapshot are forwarded and stored on a server. The uniqueness of the hash functions signed with the user terminal's private key ensures that the raw PRS snapshot is cryptographically combined with the measurement.

To verify the measurement with its raw PRS snapshot, a 3rd party can generate the same hashes out of the provided data and compare them to the transmitted hash sums. The integrity of the provided hash sums can be verified using the user's device published public key. [1]

*Communication Channel*

Depending on the actual application, a uni-directional or bi-directional communication link might be used.

The easiest kind of uni-directional link is to store all snapshots on a local storage volume inside the user terminal and to forward the raw data snapshots only to the PRS remote processing server when convenient. In general, also mobile communication links can be used for it. The uni-directional approach does not provide real-time tracking functionality but still provides the possibility to check or prove that a georeferenced action or time has taken place at a PRS guaranteed location and time.

Using the bi-directional link to forward the PRS snapshot with meta data to the PRS remote processing server, the server can calculate a snapshot PVT using the raw samples and send this information back to the user terminal. Depending on the application, communication channel used, snapshot-length, assisted data and so on, a certain latency has to be taken into account.

It is crucial for all mobile communication links to reduce the snapshot size as far as possible, because bandwidth is generally limited and expensive. Therefore, the impact of possible compressions are discussed in section "Data Compression Effects". However, experiments with Universal Mobile Telecommunications System (UMTS) communication link modules have shown that the actual mobile network login requires much more power than transmitting several hundred kBytes of raw data [4]. Therefore, an excessive raw size reduction might not even be needed. But again, such statements very much depend on the actual application and technology used to realize the communication link.

*OS/PRS Remote Processing Server*

First, the server has to authenticate the user terminal to ensure that only registered devices are allowed to use its re-

mote OS/PRS processing service. Then, the server can use the full OS/PRS processing capabilities in a secure environment to calculate the position, velocity and time (PVT) solution on basis of the PRS. In order to do so, the snapshot algorithm block is connected to a PRN code provider. For PRS the PRN code provider is crucial, since it includes a PRS security modules with the protected and classified cryptographic algorithms to generate the PRS PRN streams. The actual snapshot positioning algorithms used are outlined in the following section.

## SNAPSHOT ALGORITHMS

A conventional receiver acquires and tracks the available Global Navigation Satellite Systems (GNSS) signals and obtains all the necessary navigation data during its tracking phase. The snapshot remote processing server has to cope with only a few milliseconds of recorded data. Therefore, a conventional tracking approach is not possible.

Since the broadcasted ephemeris data and correction parameters cannot be decoded from the short raw data snapshot, a secondary channel has to be used as an external data source.

For a OS based snapshot receiver, two further challenges exist: the ambiguities in the pseudoranges and the coarse time error. Both challenges do not exist for a PRS snapshot receiver, which has some other difficulty to resolve.

### *Pseudorange Reconstruction based on a priori Position and Time*

The pseudorange computation, which is based on signal transmission and arrival time, becomes challenging, since the signal time of transmission broadcasted by the satellite is not available. The receiver can only detect the code phase within the signal's code period.

As an example, a Galileo OS signal has a symbol period or spreading code duration $\tau_{E1BC}$ of 4 ms, which is about 1200 km in the spatial domain. So the known fractional part of the pseudorange $P_{E1BC}$ is restricted to the last 1200 km. Since the traveling time of the signal from satellite to receiver will be approximately 20 to 25 times higher than the spreading code duration, an ambiguity term needs to be solved.

Van Diggelen [5] proposes a pseudorange reconstruction algorithm to resolve the code period integer ambiguity for each satellite based on an *a priori* time of transmission $t_{ref}$ ($X_{SV}(t_{ref})$) and user position $X_{RX}$ given within a certain accuracy. With available ephemeris, one can compute the expected geometric range $\rho$ and solve the common bias $b_{ref}$ term by setting an arbitrarily chosen integer ambiguity $N_{ref}$ for a reference satellite ($SV_{ref}$)

$$b_{ref} = N_{ref} \cdot \tau_{E1BC} + P_{E1BC,SV_{ref}} - (\rho(X_{RX}, X_{SV_{ref}}(t_{ref})) - \delta t_{clk,SV_{ref}}) - \epsilon_{SV_{ref}}, \quad (2)$$

where $\epsilon_{SV}$ is the satellite dependent range error and $\delta t_{clk,SV}$ the satellite clock error. With the computed common bias $b_{ref}$ for the reference satellite the ambiguity for every other acquired satellite can be solved as

$$N_{SV} = \left\lfloor \frac{\rho(X_{RX}, X_{SV}(t_{ref})) - \delta t_{clk,SV} + b_{ref} + \epsilon_{SV} - P_{E1BC,SV}}{\tau_{E1BC}} \right\rceil, \quad (3)$$

where $N_{SV}$ is rounded to the nearest code duration integer (i.e. 1 unit ms for GPS L1 C/A and 4 ms for Galileo E1 OS) as indicated by $\lfloor \cdot \rceil$.

If the *a priori* time of transmission and user position have been accurate enough, all the ambiguity terms are solved correctly and the so called full pseudorange can be reconstructed. As stated for GPS L1 C/A signal in [5], accurate enough means that the time of transmission may possess an error margin of up to 1 min and user position of up to 150 km.

### *Pilot Secondary Code Pseudorange Reconstruction*

Instead of using the pseudorange ambiguity resolution technique described before, we propose a different approach by using the GNSS pilot signals' secondary code (where available) to directly obtain an unambiguous pseudorange measurement.

Modern GNSS have pilot signals, like the Galileo E1C, E5aQ and E5bQ and GPS L5Q, using a secondary code on top of their primary spreading code to generate a long, so called *tiered* code. The overall code duration is the product of primary and secondary code lengths.

For the Galileo pilot signals E1C, E5aQ and E5bQ, the overall tiered code period is 100 ms. A tiered code length of 100 ms can resolve distances of approx. 30.000 km in an unambiguous way which is enough for a direct measurement of the distance between satellite and receiver.

However, in the standard acquisition method, one only acquires a single primary code sequence to obtain the ambiguous pseudorange measurement. Therefore, one does not directly know at which point in the secondary code the measurement was taken leading to ambiguity one more. To solve it we utilize a search method exploiting the fact that the secondary code sequences are a priori known.

The secondary code acquisition algorithm reconstructs both the code phase offset and the secondary code. This enables the direct computation of full pseudorange. In the case of the secondary code acquisition, full pseudorange computation is not based on the prediction of the user position. One advantage of the procedure is that no matter what a priori user position was given, the position converges to the actual, exact user position.

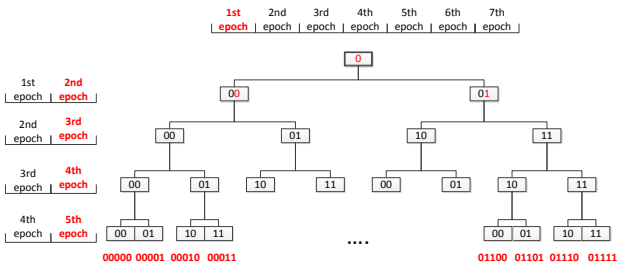



**Figure 3.** Secondary code acquisition procedure binary search tree scheme

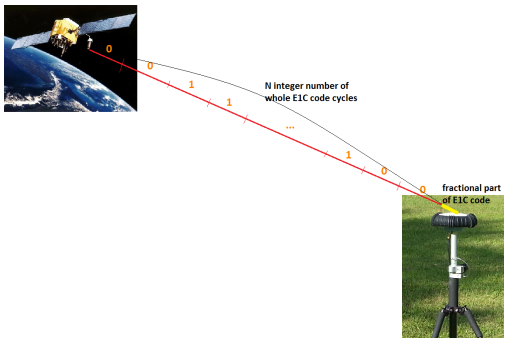



**Figure 4.** Secondary Code Pseudorange Reconstruction

Figure 3 shows a secondary code reconstruction binary tree. In each step, two coherent correlations are done using two consecutive replica symbols: the first correlation is done with both symbols having the same phase (e.g. [0 0]), the second correlation with symbols having the opposite phase (e.g. [0 1]). The two resulting correlation peaks are compared and the higher one (typical approx. twice as high) is identified as the correct estimation. The height of this binary search tree depends on the properties of the reconstructed secondary code. There is no need to go through all nodes. Instead, the decision for the right branch is made in each step.

Finally, the computed estimated secondary code phases are compared with the known secondary code. This stage is illustrated in Figure 4 for E1C, where the pilot code's secondary code symbols 0, 0, 1, 1, ..., 1, 0, 0 have been sent while the user received part of the primary code modulated with secondary code value 1 and further secondary code values: 0, 0, 0, 1, 1. Since the secondary code is predefined and repeatable resulting in 00111000000010101101100**10 0011**... So, the 23 whole symbols passed before the primary code fractional part has been received. This way, the full pseudorange can be directly obtained.

In general, in remote server applications, both the execution time and computational power are not that relevant. Therefore, more hypothesis may be covered simultaneously, as mentioned in [6] and [7]. The same facts also support a possible longer correlation time. However, after tens of milliseconds the user terminal's oscillator stability sets the limit. Although a longer snapshot is needed for this method to work, the advantages might be worth considering it.

*Coarse Time Error Solution*

Since a snapshot is only a few milliseconds long, it is impossible to obtain the signals' time of transmission through message decoding. Therefore, the assumed time of transmission is only an estimate and the resulting time error, called coarse time error $\delta_{tc}$, causes the satellite positions used in the positioning least squares algorithm to be wrong.

Therefore, the common pseudorange observation equation needs to be extended with the product of coarse time error and pseudorange rate.

$$P_{E1BC,SV} + N_{SV} \cdot \tau_{E1BC} = (\rho(X_{RX}, X_{SV}(t_{ref})) - \delta t_{clk,SV}) + b_{ref} + \dot{\rho}(X_{RX}, X_{SV}(t_{ref})) \cdot \delta_{tc} + \epsilon_{SV} \quad (4)$$

Pseudorange rate information is available through either Doppler measurements or the satellite velocities given in the ephemeris. By extending the state of unknowns, it is necessary to receive and acquire at least five satellite signals for three dimensional (3-D) positioning. The state is given through the unknown receiver position, the common bias and the coarse time error

$$state = [X_{RX}, Y_{RX}, Z_{RX}, b_{ref}, \delta_{tc}]^T . \quad (5)$$

*Snapshot Positioning for PRS*

PRS signals are non-periodic, therefore, a pseudorange reconstruction to solve the ambiguity is not necessary, since the PRS's code length exceeds the satellite signal's propagation time.

Another advantage is that the PRS PRN sequences are valid only for a specified time period. Once the matching between the snapshot and this sequence is found, the time of transmission for this particular satellite can be determined. Therefore, one does not need an a priori time of transmission estimate for PRS and other non-repeating PRN signals. The more complex snapshot positioning algorithm with five unknowns, including the coarse time error, can be reduced to the common four unknown positioning algorithm.

However, snapshot positioning with Galileo PRS brings up a challenge for the acquisition process. Since the PRS's spreading code is non-periodic, precise time information is required to perform acquisition within a certain predefined time frame, otherwise the computational complexity increases. To do so, either a dedicated direct PRS acquisition module has to be used, or time has to be obtained from the OS components (thus, an OS-assisted-PRS acquisition), or transmitted via a secondary channel (e.g. having a stable real-time clock in the user terminal). Any time uncertainty leads to a direct increase of the code phase search space.

## DATA COMPRESSION EFFECTS

A conventional GNSS receiver first acquires the signal to estimate its coarse code phase and Doppler. Then the code phase and Doppler frequency are refined within the tracking process.

In contrast to such a conventional GNSS receiver, a snapshot based receiver cannot track the GNSS signals, since the snapshot length is normally much too short for that. Therefore, the remote server can only use an acquisition to estimate the code phases and pseudoranges of the signals for the following PVT calculation. The direct estimation of the code phase and Doppler, without refinement in a tracking stage, can add some addition signal-to-noise ratio (SNR) degradation as discussed in the following.

For a successful and accurate snapshot position two problems have to be solved: Firstly, a detection problem of the GNSS signals buried in noise, secondly, the accurate, instant estimation of the code phase to derive the pseudorange measurement. Both problems are linked to the snapshot data size which is the product of the complex sampling rate $f_s$, quantization bits $Q$, and snapshot length $l$, see Eq. 1.

The reduction of the sampling rate $f_s$ impacts both the probability of detection as well as the accuracy, as outlined in the following, whereas the reduction of the quantization $Q$ and the snapshot length $l$ only impacts the detection probability.

*Probability of Detection*

In snapshot positioning, the only measurement for the signal detection and pseudorange determination is the acquisition detector's output. The detector output $D$ can be defined as

$$D = \sum_{k=1}^{N_{NI}} 2T_{int} \frac{C}{N_0} R^2(\Delta\tau) \mathrm{sinc}^2\left(\pi \Delta f_D T_{int}\right) , \quad (6)$$

ignoring the presence of navigation bit transitions. $N_{NI}$ is the number of incoherent integrations, $T_{int}$ denoting the coherent integration time, $C/N_0$ being the received signal's carrier to noise density, $R(\Delta\tau)$ being the cross ambiguity function (CAF) depending on the code phase $\Delta\tau$, and $\Delta f_D$ being the carrier frequency error.

Using the signal detection probability [8], one can define the probability of detection using the detector's output $D$ from Equ. 6 and a given false alarm rate $\alpha$. Figure 5 shows the detection rate for a given $\alpha$ of 0.1 % on the dependency of $T_{int}$, $N_{NI} = 1$ and $C/N_0$. Increasing any of those parameters results in a higher detection rate.

The $C/N_0$ is affected by the snapshot receiver's parameters discussed in the following. These additional losses lead to

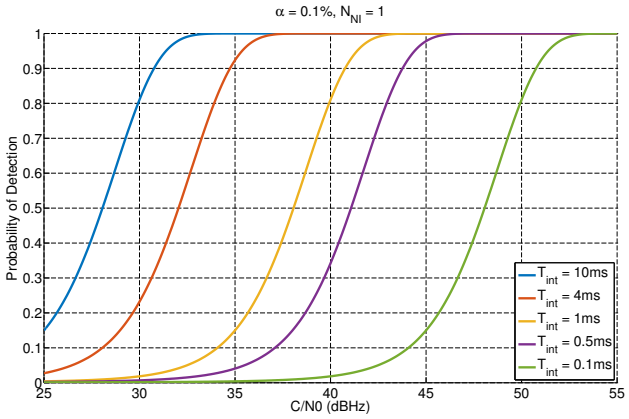



**Figure 5.** Probability of detection for a fixed probability of false alarm and different coherent integration times $T_{int}$

a lower $(C/N_0)_{eff}$ than for conventional receivers. With a lower $(C/N_0)_{eff}$ through compression losses, $N_{NI}$ or $T_{int}$ can be increased by the user to still acquire enough satellites for calculating a position solution at the server, at the expense of a longer snapshot time and therefore size.

*Degradation Effects*

In order to minimize the transmission data between client and server, the reduction of the captured raw data at the snapshot receiver is required.

Applying this data compression on the received GNSS signal leads to degradations in SNR and position accuracy. The effective carrier-to-noise density $(C/N_0)_{eff}$ is affected by several signal losses due to filtering, quantization, and acquisition losses, as well as some antenna/receiver specific degradations

$$(C/N_0)_{eff} = C/N_0 - L_B - L_Q - L_A - L_{Rx} , \quad (7)$$

where $C/N_0$ represents the nominal carrier-to-noise density of a receiver under ideal reception conditions without any SNR degradation effects. The losses due to band-limiting/filtering of the signals are expressed with $L_B$, the quantization loss with $L_Q$, the acquisition introduces losses with $L_A$, and finally, the antenna/receiver specific losses with $L_{Rx}$.

In [9] these effects have been thoroughly analyzed and will be summarized in the following.

*$L_B$: SNR Degradation due to Band-limitation/Filtering*

The Galileo PRS signals can be processed either using a correlation with the full BOCc modulated replica ("full-BOC" method), depicted in Fig. 6 or by correlation with only a replica of the single sideband (SSB) ("SSB method") depicted in Fig. 7.

Using numerical simulations to assess the effect of a brick-wall band limiting filter on the E1A/E6A PSD of the BOCc

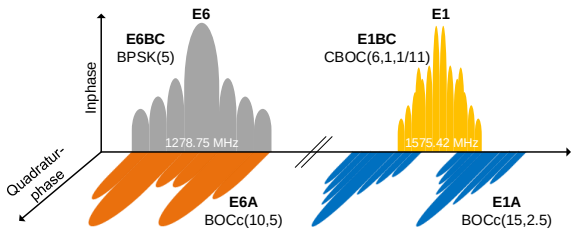



**Figure 6.** Selected frequency bands of for full-BOC processing: E1A (red), E6A (blue) PRS BOCc and E1BC OS CBOC (yellow)

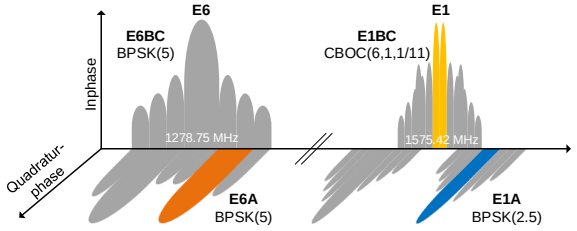



**Figure 7.** Selected frequency bands for upper sidelobe SSB processing of E1A and E6A PRS signals

signals with an assumed transmitted bandwidth of 50 MHz, it can be shown that the bandwidth limitation of 33 MHz for E1A and 25 MHz for E6A, respectively, causes a signal loss of approximately 1 dB. Further limitation of the bandwidth results in an undesired cut off of the main lobe containing most signal power.

An alternative method is the SSB BPSK, in which only the upper or lower sidelobe is processed as depicted in Fig. 7. Here, the modulation of E1A and E6A is approximately a BPSK(2.5) and BPSK(5) signal, respectively. Using only a single sideband adds a loss of approx 3 dB to the $C/N_0$, but the necessary bandwidth (and therefore sampling rate $f_s$) is much smaller than for the full-BOC method.

Targeting again a band limiting signal loss of less than 1 dB using an assumed transmitted bandwidth of 20 MHz, the band-limitation should be higher than 3 MHz and 6 MHz for E1A and E6A, respectively.

*$L_Q$: SNR Degradation due to Quantization*

The quantization defines the overall dynamic that can be used in the digital signal processing with approximately 6 dB per analog-to-digital converter (ADC) quantization bit.

If the sampling rate is much higher than the signal's chipping rate, the SNR degradation becomes negligible with more than 3 appropriately controlled quantization bits. Under the assumption of an ideal gain-controlled ADC, for 1, 2 and 3 bits the resulting SNR degradation is 1.96 dB, 0.55 dB and 0.17 dB, respectively. If the sampling rate is approx. only twice the signal's chipping rate, the losses are approx. 0.5 dB higher [10].

*$L_A$: SNR Degradation due to Acquisition/Sampling Constraints*

In the acquisition process, the code phase (i.e. delay or time measurement of the signal) as well as the Doppler frequency are estimated by the evaluation of the correlation or CAF. The additional losses in this processing stage can occur from code sampling as well as from frequency estimation mismatch. These losses are also named scalloping losses [11] and can be expressed with:

$$L_A = L_{code} \cdot L_{Doppler} \tag{8}$$

The received signal is band-limited to be able to reduce the sampling rate $f_s$ and, in the same way, the snapshot's data size. According to the Nyquist theorem, the complex baseband bandwidth of $[-B_F/2, +B_F/2]$ is limited by the Nyquist sampling rate ($f_s \geq B_F$). A lower sampling rate leads to aliasing effects, e.g. folding noise into the useful band and degrading the SNR.

The reduced sampling rate/bandwidth can, on top of the filtering losses, lead to additional SNR degradation due to the fact that the sampling of the receiver is not synchronized with the transmitted signal.

The $L_{code}$ depends on receiver's sampling points $N_s$ on the correlation function of the processed GNSS signal $R(\Delta\tau)$:

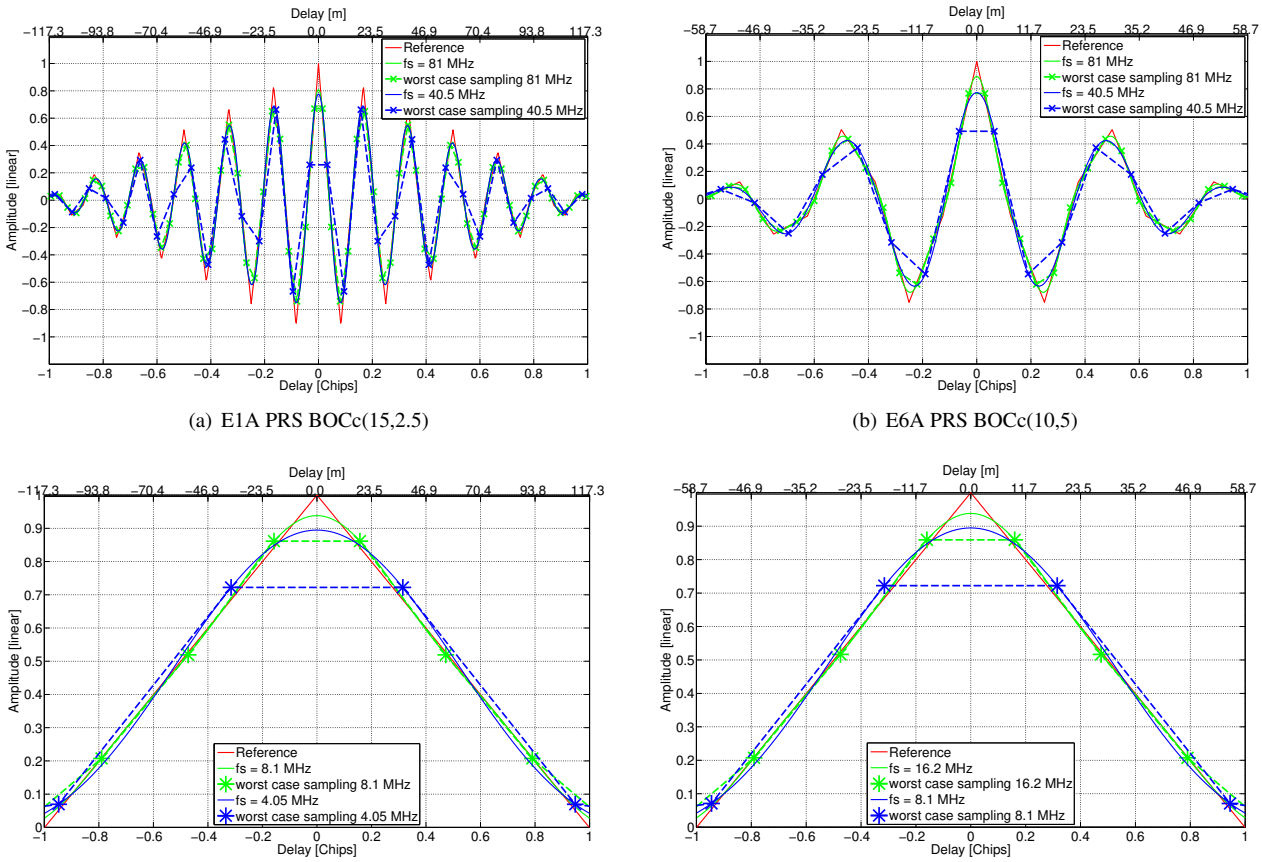
$$L_{code} = R(\Delta\tau - N_s) \tag{9}$$

It represents the difference between the bandlimited correlation peak and the "off-grid" CAF point due to the unsynchronized CAF sampling process.

Figures 8 show "worst case" code mismatch scenarios for the PRS signal E1A BOCc(15,2.5) and E6A BOCc(10,5) processed in full BOC and SSB modes. These scenarios are regarded as "worst case" since their sampling points are exactly half the sampling interval off the main peak, leading to the biggest $L_{code}$ SNR loss.

Starting from an arbitrarily chosen baseline sampling rate of 81 MHz, the sampling rates are halved to 40.5 MHz for the full-BOC PRS signal processing, to still be able to receive the full-BOCc PRS signal. Complex sampling (i.e. both in-phase and quadrature-phase simultaneously) is assumed and an ideal "brick wall" band limiting filter equal to the complex sampling rate is applied.

As depicted in Figure 8(a), for the E1A PRS BOCc(15,2.5) with 40.5 MHz the sample on the main peak has a worst case code mismatch loss of 9.5 dB with respect to the rounded main peak (also respecting an assumed transmission bandwidth of 50 MHz). In this case, the side peak with only a loss of 1.3 dB would be acquired. Besides the SNR losses, these worst case sampling points would add a bias of 3.8 m




(a) E1A PRS BOCc(15,2.5)

(b) E6A PRS BOCc(10,5)

(c) E1A PRS single sideband BPSK(2.5)

(d) E6A PRS single sideband BPSK(5)

**Figure 8.** "Worst case" code phase sampling mismatch on CAF for different sampling rates and E1A/E6B processing in full-BOC and SSB

and 11.1 m for the worst case sampling on the main and first sidepeak, respectively.

The situation depicted in Figure 8(b) is similar for E6A PRS BOCc(10,5) with 40.5 MHz. Here, the worst case loss w.r.t. the rounded main peak respecting an assumed transmission bandwidth of 50 MHz is 4.2 dB, whereas the worst case loss on the second peak is 3.3 dB that could still lead to a false peak acquisition. This also would end up in the same worst case pseudorange biases as reported in the E1A case.

To mitigate this part of the scalloping loss and acquisition of a false BOC peak, a sideband processing technique of the BOC signals can be applied: instead of the multi-peak BOCc signals, only its BPSK counterparts without any side peaks have to be acquired.

SSB processing of the BOCc modulated PRS signals is very attractive for a snapshot receiver: Firstly, the much lower sampling rate leads to smaller raw data snapshot sizes. Secondly, the BPSK modulations only have one single peak, prohibiting the likely false peak locks in the full BOC processing. However, using only a single sideband results in a 3 dB SNR loss, which can be mitigated by a non-coherent combination of both lower and upper side-lobes, increasing the snapshot size again.

As depicted in Figure 8(c), using a sampling rate of only 4.05 MHz, the BPSK(2.5) SSB processing of E1A leads to a worst case loss of 1.9 dB. However, the code phase bias is 37.0 m, even though there is no false lock possibility anymore.

For the BPSK(5) SSB processing of E6A as depicted in Figure 8(d), a worst case loss of 1.9 dB is possible when a sampling rate of 8.1 MHz is used, also leading to a code phase bias of 18.5 m.

To better estimate the true code phase, interpolation techniques, e.g. using a spline interpolation between the CAF sampling point, can mitigate these significant worst case code phase offsets for very low sampling rates.

The $L_{Doppler}$ depends only on the Doppler search grid of the acquisition stage with its Doppler frequency error $\Delta f_D$ and the chosen coherent integration time $T_{int}$:

$$L_{Doppler} = \mathrm{sinc}(\pi \, T_{int} \, \Delta f_D) \tag{10}$$

As the coherent integration time increases, this effect becomes more and more severe. This loss can be mitigated

by selecting a fine enough Doppler grid spacing $\Delta f_D$ leading to an increased computational complexity on the server processing side.

*$L_{Rx}$: SNR Degradation due to Antenna/Receiver Constraints*

Finally, there are some antenna and receiver specific SNR degradation losses which do not depend on the data compression.

A suitable antenna to receive the wideband PRS signals is required. Most GNSS signals, including the Galileo OS and PRS signals, are transmitted right hand circular polarized (RHCP). Consequently, a RHCP antenna should be used at the user terminal's receiver. However, cheap and often small antennas, e.g. the typical patch antennas, are only linearly polarized, leading to an additional polarization SNR degradation of 3 dB. Miniaturized antennas also do not have a high efficiency or gain, which can lead to a few dBs of additional SNR degradation. Lastly, the antenna reception bandwidth, including its RF filters, can add the same SNR degradation to the signal as discussed in the bandwidth section of the data compression.

The noise figure (NF) of the data grabber (i.e. the antenna with receiver front-end) directly adds up to the SNR degradation. If an active antenna is used, the overall NF is dominated by the first low-noise amplifier (LNA) of the active antenna according to Friis' equation. A typical overall NF is approximately 1.5 to 2 dB. In case of a passive antenna, the overall NF can increase considerably, depending on the antenna efficiency with the cable and the LNA of the receiver reaching 5 to 7 dB or even more.

## EXPERIMENTAL SETUP AND RESULTS

*Test Setup*

An empirical analysis has been conducted for PRS snapshot position solutions using Galileo E1A Pseudo-PRS signals. Since the Galileo constellation is not yet complete and Pseudo-PRS instead of real PRS signals shall be used to keep the experimental setup unclassified, a Spirent GSS9000 radio frequency constellation simulator (RFCS) is used to simulate a full Galileo constellation scenario.

The Spirent GSS9000 can substitute the real PRS signals by Pseudo-PRS or PRS-noise signals, where the E1A / E6A PRS modulations of BOCc(15,2.5) and BOCc(10,5), respectively, are preserved, but the PRN chips are generated using the unclassified GPS P-Code. The performance in terms of accuracy is equivalent to the real PRS signals, since the accuracy is only dependent on the modulation, scenario and data grabber emulation model, which is the same for both PRS and Pseudo-PRS.

For our experiments, we simulated a static scenario (position 0 °/0 °) without atmospheric effects in Spirent's SimGEN default Galileo constellation. The simulated satellites

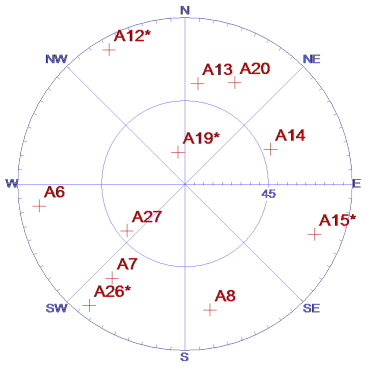



**Figure 9.** Satellite constellation with GDOP = 1.56

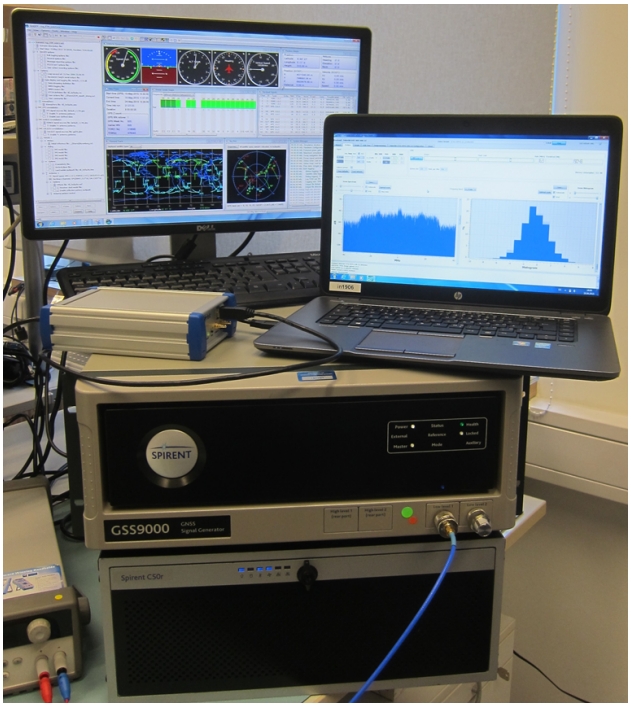



**Figure 10.** Experimental setup consisting of a RFCS and a USB front-end recording device

have a $C/N_0$ of approximately 47 to 49 dBHz, depending on their elevation. The Galileo satellite constellation with a geometric dilution of precision (GDOP) of 1.56 is illustrated in Fig. 9.

The RFCSs output signals are received by a front-end and saved to hard disk for the following processing modules. To be able to process the wideband BOCc(15,2.5) modulated signals, the Flexiband USB3.0 front-end of Fraunhofer [12] is used in configuration with 81 MHz sampling rate at 4 bit I/Q. The experimental setup is shown in Fig. 10.

*Emulation Model*

For a statistical analysis an OS/Pseudo-PRS signal with a duration of a few second was recorded from the RFCS. The snapshot acquisition and positioning algorithms are applied

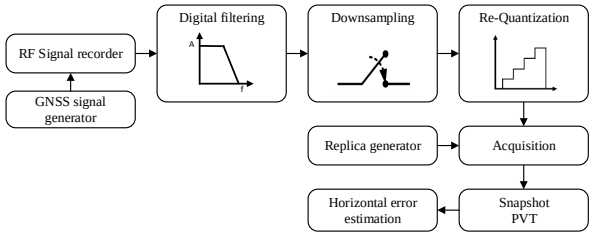



**Figure 11.** Emulation model for snapshot positioning evaluation with adjustable blocks for digital filtering, downsampling and re-quantization

on snapshots obtained by stepping through the recorded file and treating the individual sub-snapshots as completely independent. The system emulation model is illustrated in Fig. 11. As outlined above, the test setup is used to record binary raw data that include the E1B OS and E1A Pseudo-PRS signals. For each scenario, the defined parameter settings are chosen for digital filtering, downsampling and re-quantization to simulate the data compression effects. After the satellites have been detected in the acquisition, a snapshot PVT is performed on the acquisition output, in which the OS results are used to acquire the Pseudo-PRS signals. The horizontal error is calculated and compared for each experimental trial.

*Determining Combined Losses*

In order to minimize the data size for the communication link, the snapshot receiver has to accept additional SNR losses. An assumed miniaturized user terminal design includes all the additional antenna/receiver related losses discussed before. Adding the effects of data compression for a snapshot generation (band-limitation, sample rate constraints, and quantization), the resulting signal loss can be seen in Table 1 showing the calculated examples for E1A PRS with 10 ms snapshot length and the following parameters:

- Scenario 1: Large bandwidth and high sampling rate, multi-bit quantization (81 MHz, 4 bit)
- Scenario 2: Limited bandwidth and sampling rate as well as single bit quantization (40.5 MHz, 1 bit)
- Scenario 3: Single-sideband, low sampling rate as well as single bit quantization (8.1 MHz, 1 bit); with and without spline interpolation between the CAF points
- Scenario 4: Single-sideband, low sampling rate as well as single bit quantization (4.05 MHz, 1 bit); with and without spline interpolation between the CAF points

The SNR degradation due to antenna/receiver constraints $L_{Rx}$ is dominated by the used data grabber's NF only, which

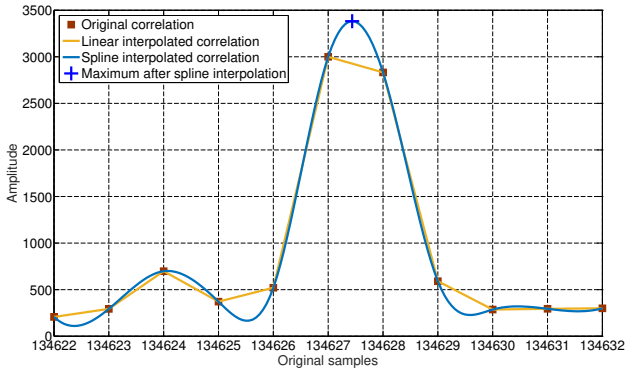



**Figure 12.** Exemplary acquisition result of Scenario 4: original sampling points on the CAF vs. the result of the spline interpolation for pseudorange determination

is approx. 2 dB. In reality, this value can be much higher, especially with low cost miniaturized antennas.

The SNR degradations due to acquisition/sampling constraints $L_A$ are given in a range between the best and worst case loss. For the full BOC processing, the losses for a possible false sidepeak acquisition in the worst case situation are given in brackets.

*E1A Simulation Results*

Based on the Pseudo-PRS-PRN of the Spirent's signal generator, we conducted an evaluation of E1A PRS signals with a snapshot length of 10 ms for the previously introduced four emulated scenarios with 200 trials each.

With the maximum front-end bandwidth of 81 MHz, Scenario 1 represents a setup without any constraints. From Fig. 14(a), we see a small variance of approx. 0.85 m circular error probability (CEP) and a mean of nearly zero. Similar findings are also observed for Scenario 2 in Fig. 14(b), where the bandwidth is halved and quantization bits reduced to only 1 bit resulting in less than 2 m CEP.

In Scenario 3, the sampling rate is only a tenth of the original frequency, leading to a high reduction factor of the snapshot size. In Scenario 4, the sampling rate is halved again. With approx. 4 MHz sampling rate on a BPSK(2.5) signal, only approx. 3-4 samples fall on the CAF-peak.

By considering the effect of a SSB band limiting filtering without any interpolation, the variance increases considerably as plotted in Fig. 14(c) and Fig. 14(d).

To estimate the code phase more accurately, a spline interpolation through the CAF samples is applied to approximate the true pseudorange in a more accurate way than just taking the highest sample. One example can be seen in Fig. 12. Using such an interpolation to refine the code phase estimation, the variance for Scenario 3 reduces from 9.7 m to 3.8 m $1\sigma$ and for Scenario 4 from 19.2 m to 6.0 m, respectively, as shown in Fig. 14(e) and 14(f).

**Table 1.** Comparison of signal losses and data size for the four emulated E1A (Pseudo-)PRS scenarios

| **Data compression effects** | **Scenario 1** full BOCc(15,2.5) | **Scenario 2** full BOCc(15,2.5) | **Scenario 3** SSB BPSK(2.5) | **Scenario 4** SSB BPSK(2.5) |
|---|---|---|---|---|
| $f_s$ [MHz] | 81.00 | 40.50 | 8.10 | 4.05 |
| $Q$ I/Q [bit] | $2 \times 4$ | $2 \times 1$ | $2 \times 1$ | $2 \times 1$ |
| $l$ [ms] | 10 | 10 | 10 | 10 |
| **Data size [kByte]** | 810.000 | 101.250 | 20.250 | 10.125 |
| $L_B$ [dB] | 0 | 0.4 | 3.3 | 3.7 |
| $L_Q$ [dB] | 0 | 2.4 | 2.4 | 2.4 |
| $L_A$ [dB] | 0...1.6 (0.8) | 0...9.5 (1.3) | 0...0.7 | 0...1.9 |
| $L_{Rx}$ [dB] | 2 | 2 | 2 | 2 |
| **overall SNR loss [dB]** | 2...3.6 (2.8) | 4.8 ... 14.3 (6.1) | 7.7 ... 8.4 | 8.1 ... 10.0 |

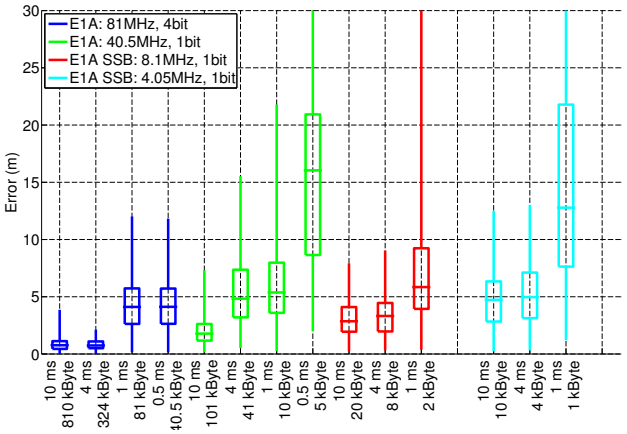



**Figure 13.** Comparison of horizontal errors for the four compression scenarios varying the snapshot length with its corresponding sizes; the boxes indicate the 50 % percentile with the median value inside

Figure 13 reuses the four defined scenarios but now reduces the snapshot size—and therefore the coherent integration time $T_{int}$—to 10 ms, 4 ms, 1 ms and 0.5 ms. With the reduction of the coherent integration time $T_{int}$, the probability of detection decreases accordingly, as shown in Fig. 5. Starting from the initially simulated approx. 48 dBHz, one has to decrease the compression losses accordingly. To this end, very short integration times do not provide enough successfully acquired satellites leading to very big errors in the position or rendering position calculation entirely impossible, since less than the required 4 satellites are detectable. However, this simulation demonstrates, that in principle, a PRS raw data snapshot of 1 kByte can be enough to calculate a secure, unspoofable and trustable PRS snapshot PVT.

## ACCURACY ESTIMATION

In most snapshot applications, only a single snapshot PVT solution is available. However, for many applications, it is important to get an indicator of the likelihood that the true PVT position / time information is within a certain range for a certain probability.

In conventional tracking based receivers, techniques like RAIM (Receiver Autonomous Integrity Monitoring) are used for this purpose. However, in a single snapshot PVT, no statistical distribution of the pseudorange or position information is available. The available information is: residuals from the PVT calculation (assuming the system is overdetermined), an estimation of the $C/N_0$ for the acquired satellites (derived from e.g. the main peak to the second highest next peak ratio outside the CAF) and information about the GNSS from a secondary channel (e.g. SISA (Galileo's Signal-In-Space Accuracy) values).

The following method, which is similar to the RAIM-like approaches reported in [13], takes the available single snapshot PVT information into account and derives a statistical error estimate.

### *State Error Estimation*

The snapshot processing estimates the state of unknowns, containing the user position, satellite common bias and coarse time error, as the least squares solution of a linear system of equations

$$Hx = \rho \tag{11}$$

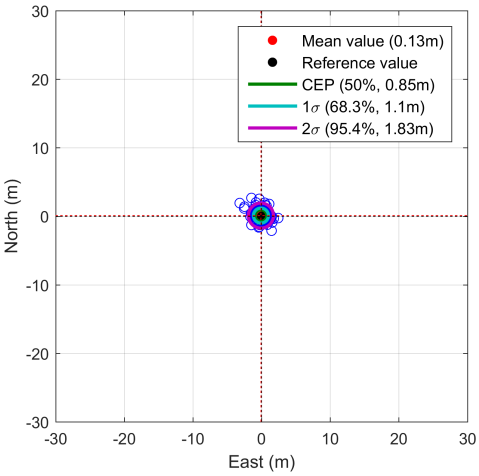
Here $H$ is the measurement matrix, containing the linearized observation equations, $\rho$ are the measurements and $x$ is the computed state.

Since the measurements are corrupted with errors, the computed state is error affected as well. Therefore the system has the form:

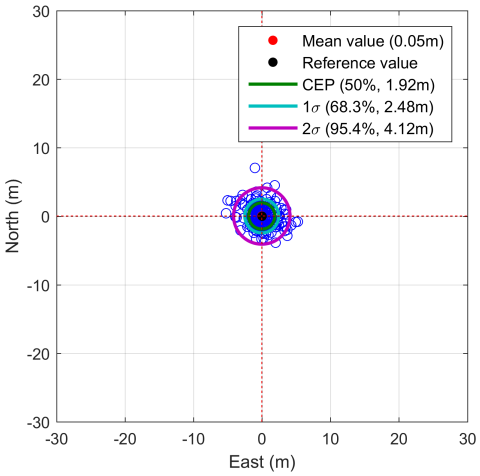
$$H(x + dx) = \rho + d\rho \tag{12}$$

where $dx$ is the error of the computed state and $d\rho$ is the error in the observed and reconstructed pseudoranges. Using a weighted least squares approach, the resulting system is $WH(x + dx) = W(\rho + d\rho)$, with $W$ being the weighting matrix. The error in the weighted least squares solution is

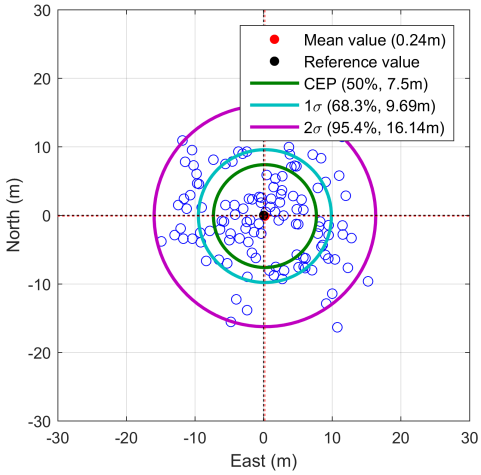
$$dx = (H^T W H)^{-1} H^T W d\rho. \tag{13}$$



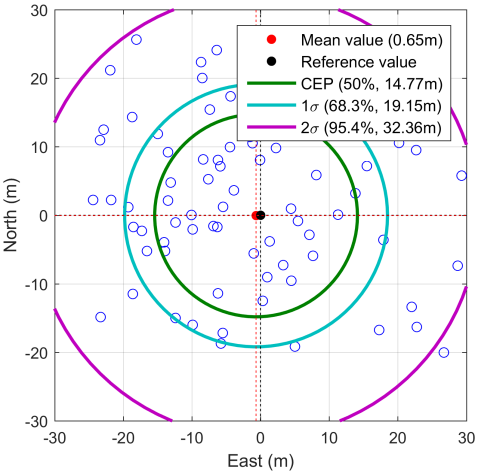

(a) Scenario 1: 81 MHz sampling rate, 4 bit I/Q, BOCc(15,2.5) processing



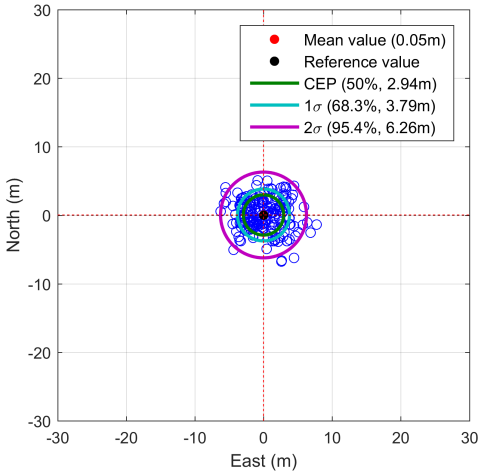

(b) Scenario 2: 40.5 MHz sampling rate, BOCc(15,2.5) processing



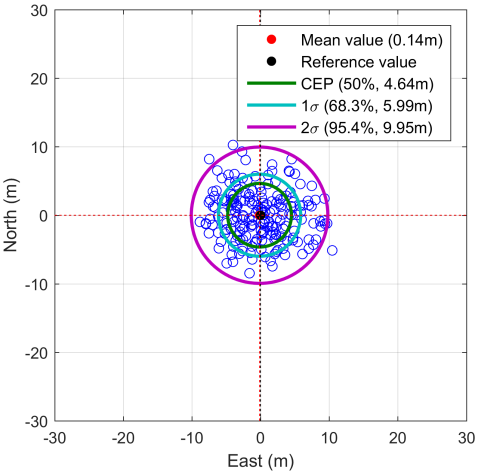

(c) Scenario 3: 8.1 MHz sampling rate, BPSK(2.5) single sideband processing without interpolation




(d) Scenario 4: 4.05 MHz sampling rate, BPSK(2.5) single sideband processing without interpolation




(e) Scenario 3: 8.1 MHz sampling rate, BPSK(2.5) single sideband processing with spline interpolation for code phase evaluation




(f) Scenario 4: 4.05 MHz sampling rate, BPSK(2.5) single sideband processing with spline interpolation for code phase evaluation

**Figure 14.** Snapshot positioning horizontal error for different scenarios of E1A Pseudo-PRS

Its corresponding state error covariance matrix is:

$$cov(dx) = \mathbb{E}[dxdx^T] \tag{14}$$
$$= \mathbb{E}[(H^TWH)^{-1}H^TWd\rho d\rho^TWH(H^TWH)^{-1}] \tag{15}$$
$$= (H^TWH)^{-1}H^TWcov(d\rho)WH(H^TWH)^{-1} \tag{16}$$

Based on the previous computation, the corresponding choice of the weighting matrix would be

$$W = cov(d\rho)^{-1} = \begin{bmatrix} \frac{1}{\sigma_1^2} & 0 & \cdots \\ 0 & \ddots & \ddots \\ \vdots & \ddots & \frac{1}{\sigma_N^2} \end{bmatrix} \tag{17}$$

with $\sigma_i^2$ being the range measurements' variances consisting of

$$\sigma_i^2 = \sigma_{SIS_i}^2 + \sigma_{RX_i}^2. \tag{18}$$

$\sigma_{SIS_i}^2$ is the satellite error reported in the navigation message (e.g. for GPS the User Range Accuracy (URA) and for Galileo the SISA (Signal-in-Space Accuracy)) together with remaining errors from the atmosphere.

$\sigma_{RX_i}^2$ is the result of firstly, the common receiver range error due to noise, secondly, the influence of received signal's $C/N_0$ and thirdly, the range error resulting from non-optimal estimation of the acquisition function's code phase (e.g. coming from limited number of samples / interpolation) including potential multipath.

With this weighting approach, the state error covariance matrix is:

$$cov(dx) = (H^TWH)^{-1} \tag{19}$$
$$= \begin{bmatrix} \sigma_{xx}^2 & \sigma_{xy}^2 & \sigma_{xz}^2 & \sigma_{xb}^2 & \sigma_{xt_c}^2 \\ \sigma_{yx}^2 & \sigma_{yy}^2 & \sigma_{yz}^2 & \sigma_{yb}^2 & \sigma_{yt_c}^2 \\ \sigma_{zx}^2 & \sigma_{zy}^2 & \sigma_{zz}^2 & \sigma_{zb}^2 & \sigma_{zt_c}^2 \\ \sigma_{bx}^2 & \sigma_{by}^2 & \sigma_{bz}^2 & \sigma_{bb}^2 & \sigma_{bt_c}^2 \\ \sigma_{t_cx}^2 & \sigma_{t_cy}^2 & \sigma_{t_cz}^2 & \sigma_{t_cb}^2 & \sigma_{t_ct_c}^2 \end{bmatrix} \tag{20}$$

This covariance matrix determines the estimated position error circle similar as in the case of DOP and URA/SISA values, as shown in [13]. The distance root mean square error (RMSE) is defined as

$$CEP_{68} = RMSE = \sqrt{\sigma_{xx}^2 + \sigma_{yy}^2} \tag{21}$$

where CEP is Circular Error Probability. Similarly, all other $CEP_x$ values can be formed using the method presented in [14].

**Table 2.** Different CEP values for E1A (Pseudo-)PRS Scenario 4

| | Horizontal error success rate |
|---|---|
| $CEP_{50}$ | 55.5% |
| $CEP_{68}$ | 78% |
| $CEP_{95}$ | 98.5% |

*Proof of Concept*

To prove that this accuracy estimation method works, the same snapshots from Scenario 4 are used including the spline interpolation.

The experimental section's recording are without any atmospheric, multipath or satellite range error effects, leading to a zero value of $\sigma_{SIS_i}^2$.

For the $\sigma_{Rx}^2$, a value corresponding to the $1\sigma$ position error of Fig. 14(f) was set. The $\sigma_{RX_i}^2$ for the individual range errors is then scaled according to the different received $C/N_0$ levels. However, it should be noted that for future work, the determination of $\sigma_{Rx}^2$ needs further investigation.

The horizontal error estimation circles are plotted together with corresponding position errors and coordinate origin as the reference.

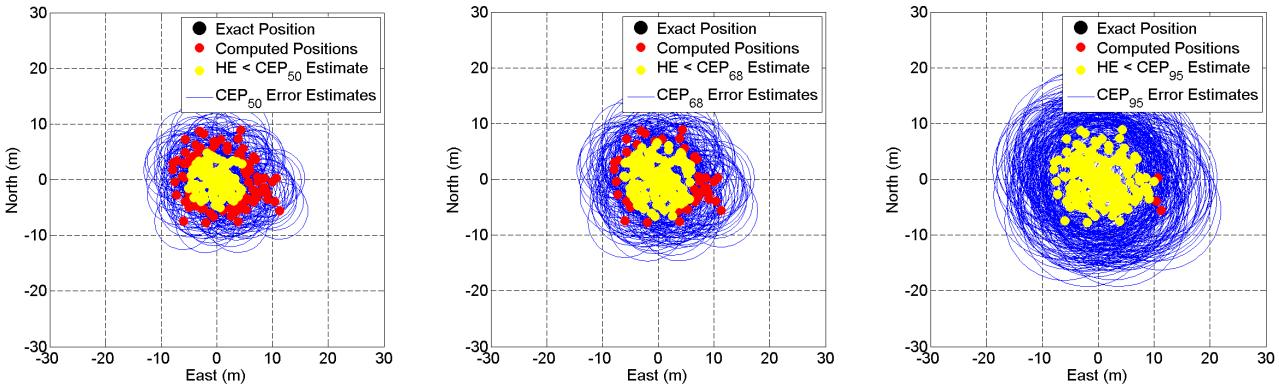
Figures 15 give an overview of the estimations done using the presented method. Figure 15(a) shows the CEP error estimation results when using the previously proposed method. Figures 15(b) and 15(c) show $1\sigma$ and $2\sigma$ error estimates, respectively.

As shown in Table 2, the results are in the range of the theoretical expectations, indicating that this method works for the simulated cases. For future work, this accuracy estimation methods will be evaluated against real-world scenarios including also SIS errors as well as multipath effects.

## CONCLUSION

The proposed OS/PRS snapshot receiver architecture with server-side processing is a novel concept allowing a greater government authorized community to potentially benefit from the unique anti-spoofing and encryption properties of Galileo PRS. Server-based technologies using sample and processing methods offer the opportunity to overcome the drawbacks of conventional PRS receivers with integrated security modules by outsourcing the security processing steps to a protected and authorized server environment.

This paper first outlined the architecture with user terminal, communication channel and remote server. Then the snapshot positioning algorithms were presented, introducing a novel method which exploits the secondary code to directly obtain an unambiguous pseudorange estimation from a single snapshot measurement, as well as the differences, challenges and advantages for snapshot processing with PRS signals.




(a) $CEP_{50}$ horizontal error accuracy (b) $CEP_{68}$ horizontal error accuracy or $1\sigma$ (c) $CEP_{95}$ horizontal error accuracy or $2\sigma$

**Figure 15.** Snapshot positioning accuracy estimation for E1A Pseudo-PRS scenario 4: 4.05 MHz sampling rate, BPSK(2.5) single sideband processing with spline interpolation for code phase evaluation

To reduce the snapshot size, different data compressions like filtering/ resampling and quantization reduction are possible, however, degrading the effective $C/N_0$ and the position accuracy. The possible worst case SNR degradation of over 10 dB could lead to a decline of the probability of detection and, consequently, system performance would severely be affected.

A full BOC processing for snapshot PVT solutions is not recommended, since the sampling rate has to be very high (and so will be the snapshot size) to mitigate possible losses in the acquisition. For single sideband processing, the sampling rates and therefore the snapshot sizes can be considerably lower, however, interpolation methods have to be applied to avoid pseudorange biases and to estimate the code phase more accurately.

Using Pseudo-PRS E1 PRS RF data signals from an RF constellation simulator and applying different compression techniques, raw data snapshot sizes from 810 to 10 kByte are evaluated providing E1 PRS snapshot positioning results from 1.1 to 6.0 m $1\sigma$ accuracy. It has also been demonstrated that a PRS raw data snapshot size of 1 kByte can be enough to calculate a secure, unspoofable and trustable Galileo PRS snapshot PVT.

To estimate the accuracy of a single snapshot PVT, we propose a method that evaluates the covariance matrix of the weighted least square errors taking into account the pseudorange errors' standard deviation weighted with acquisition determined $C/N_0$ values.

Depending on the requirements of each application a snapshot can be generated with a particular data size in terms of positioning performance. The authentication of position and time of measurements can be used for monitoring of environmental parameters, preventing manipulation of evidence photos or skimming prevention systems for bank transfers. Moreover, if only the PRS raw data is presented in the raw data snapshot, the privacy and confidentiality of the user is preserved allowing applications in the area of privacy protected localizations (e.g. eCall systems for automotive vehicles, localization of elderly people).

Future work is to evaluate the remote PRS processing performance as well as the proposed accuracy estimation using real-world PRS signals in different environments like open-sky, suburban and urban scenarios.

## ACKNOWLEDGMENT

The PRSauth project is co-funded by the national program for research and innovation for space science of the Federal Ministry of Economics and Technology (BMWi). This project is also supported by the German CPA (Competent PRS Authority) as one of the first national (German) PRS pilot projects.

## REFERENCES

[1] A. Rügamer, M. Stahl, I. Lukcin, and G. Rohmer, "Privacy Protected Localization and Authentication of Georeferenced Measurements using Galileo PRS," in *IEEE/ION Position Location and Navigation Symposium (PLANS), May 5-8, 2014, Monterey, California*, 2014.

[2] A. Rügamer, P. Neumaier, P. Sommer, F. Garzia, G. Rohmer, A. Konovaltsev, M. Sgammini, S. Caizzone, M. Meurer, J. Wendel, F. F. Schubert, and S. Baumann, "BaSE-II: A Robust and Experimental Galileo PRS Receiver Development Platform," in *Proceedings of the 27th International Technical Meeting of The Satellite Division of the Institute of Navigation, ION GNSS+ 2014, September 8-12, 2014, pp. 2579-2591, Nashville, Tennessee, USA*, September 2014.

[3] Maxim, *Universal GPS Receiver, MAX2769*. https://www.maximintegrated.com/en/products/comms/wireless-rf/MAX2769.html.

[4] Daedalus Modular, Energy-independent Tracking Systems, "http://www.iis.fraunhofer.de/en/ff/lok/proj/daedalus.html."

[5] F. S. T. Van Diggelen, *A-GPS: Assisted GPS, GNSS, and SBAS, GNSS Technology and Applications Series*. Artech House, 2009.

[6] C. Palestini, *Synchronization and detection techniques for navigation and communication systems*. PhD thesis, University of Bologna, 2010.

[7] G. E. Corazza, C. Palestini, R. Pedone, and M. Villanti, "Galileo primary code acquisition based on multi-hypothesis secondary code ambiguity elimination," in *Proceedings of 20th International Technical Meeting of the Satellite Division*, 2007.

[8] S. M. Kay, *Fundamentals of Statistical Signal Processing: Detection Theory*, vol. 2 of *Prentice-Hall Signal processing series*. Prentice Hall, 1998.

[9] D. Rubino, A. Rügamer, I. Lukcin, S. Taschke, M. Stahl, and W. Felber, "Galileo PRS Snapshot Receiver with Server-side Positioning and Time Verification," in *Proceedings of DGON POSNAV 2016, Berlin, Germany*, July 2016.

[10] C. Hegarty, "Analytical Model for GNSS Receiver Implementation Losses," in *Proceedings of the 22nd International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS 2009), Savannah, GA*, pp. 3165–3178, September 2009.

[11] J. Betz, *Engineering Satellite-Based Navigation and Timing: Global Navigation Satellite Systems, Signals, and Receivers*. Wiley, 2015.

[12] A. Rügamer, F. Förster, M. Stahl, and G. Rohmer, "Features and Applications of the Adaptable Flexiband USB3.0 Front-end," in *Proceedings of the ION GNSS+*, (Tampa, Florida, USA), pp. 8 – 12, 2014.

[13] E. D. Kaplan and C. J. Hegarty, *Understanding GPS: Principles and Applications*. Artech House, 2006.

[14] W. E. Hoover, "Algorithms for Confidence Circles and Ellipses," 1984.